SENCR: A Span Enhanced Two-Stage Network with Counterfactual Rethinking for Chinese NER

Hang Zheng[1], Qingsong Li[1], Shen Chen[1], Yuxuan Liang[2], Li Liu[1]*

[1] School of Big Data and Software Engineering, Chongqing University, China
[2] The Hong Kong University of Science and Technology (Guangzhou), China
{202124131086,202124131092,202124131089}@stu.cqu.edu.cn, yuxliang@outlook.com, dcsliuli@cqu.edu.cn

Abstract

Recently, lots of works that incorporate external lexicon information into character-level Chinese named entity recognition (NER) to overcome the lackness of natural delimiters of words, have achieved many advanced performance. However, obtaining and maintaining high-quality lexicons is costly, especially in special domains. In addition, the entity ***boundary bias*** caused by high mention coverage in some boundary characters poses a significant challenge to the generalization of NER models but receives little attention in the existing literature. To address these issues, we propose SENCR, a Span Enhanced Two-Stage Network with Counterfactual Rethinking for Chinese NER, that contains a boundary detector for boundary supervision, a convolution-based type classifier for better span representation and a counterfactual rethinking (***CR***) strategy for debiased boundary detection in inference. The proposed boundary detector and type classifier are jointly trained with the same contextual encoder and then the trained boundary detector is debiased by our proposed CR strategy without modifying any model parameters in the inference stage. Extensive experiments on four Chinese NER datasets show the effectiveness of our proposed approach.

Introduction

Named entity recognition (NER) aims to detect the span and recognize the category of named entities, such as persons, locations, and organizations, in raw sentences. NER is a fundamental task in the field of natural language processing (NLP), which is necessary for many downstream NLP applications such as relation extraction (Miwa and Bansal 2016), question answering (Mollá, Van Zaanen, and Smith 2006), and knowledge graph construction (Ji et al. 2021). Compared with English NER, Chinese NER is more challenging owing to its lackness of natural delimiters in sentences (Zhu, Wang, and Karlsson 2019).

To tackle this thorny problem, lots of lexicon-based approaches (Zhang and Yang 2018; Gui et al. 2019a; Li et al. 2020; Zhao et al. 2021, 2023) were proposed to incorporate lexicon information into character-level Chinese NER models. These approaches match the input character sequence with outer lexicon and fuse the matched words' boundary and semantic features into the model's encoder input or the character representations. However, the lexicon-based approaches are suffering two main limits. Firstly, the performance of these approaches heavily relies on the quality of the lexicon, and acquiring and maintaining a high-quality lexicon can be quite costly, especially in some special domains. Secondly, the fusing strategy needs to be carefully designed to avoid introducing external noise and biases into models. These limitations have instigated our pursuit of a lexicon-free framework while achieving comparable performance.

Prior to formulating our model, we investigated the widely used Chinese NER datasets (Weibo (Peng and Dredze 2015), MSRA (Levow 2006), Resume (Zhang and Yang 2018) and CLUENER (Xu et al. 2020)). We calculated the occurrence frequency for entity boundary characters in the training sets (see Table 1) and found that the entity boundaries are dominated by a few characters, i.e., boundary characters appearing with particularly high frequency. This high-mention coverage aligns with the character-level zipf's law in Chinese languages and will contaminate NER models' generalization ability by misleading NER models to simply memorize keywords in those high-frequency mentioned boundary characters rather than leverage semantic features (Liang and Leung 2021; Lin et al. 2020). As a result, during inference on the test set, the NER model performs better on entities with high mention frequency boundary characters but exhibits lower performance on entities with less emphasized or new boundary characters. We consider this phenomenon as ***boundary bias*** and argue that it predominantly affects the identification of entity boundaries by building spurious correlations between boundary character mentions and the ground truth labels. This arises from the fact that contextual features hold greater significance in defining entity boundaries compared to character mentions.

Previous studies (Liang and Leung 2021; Zeng et al. 2020) apply entity-level data manipulations to re-balance the entity distribution in the training set so as to improve the generalization ability of NER models. However, manipulating data at the boundary-level can significantly alter the semantic meaning of words and sentences, which may greatly undermine the effectiveness of NER models. Fortunately, counterfactual inference (Pearl 2021) is promising in tackling the above spurious correlation issue caused by the unfair

*Corresponding author

high mention coverage in some boundary characters. In this paper, we aim to teach our model a combined understanding of character mention and textual context and distinguish between the effects of the character mention and textual context through counterfactual rethinking in the inference stage:
Factual Inference: *What will the prediction be if seeing both the character mention and its textual context?*
Inference with Counterfactual Rethinking: *What will the prediction be if seeing the character mention only and had not seen the textual context?*

Datasets	$r^s_{5\%}$	$r^e_{5\%}$	$r^s_{10\%}$	$r^e_{10\%}$
Weibo	0.39	0.40	0.51	0.51
MSRA	**0.63**	0.66	**0.76**	0.78
Resume	0.61	**0.83**	0.74	**0.90**
CLUENER	**0.63**	0.66	**0.76**	0.78
Average	0.57	0.64	0.69	0.74

Table 1: Statistics of the frequency ratio (r) for entity boundary characters in training sets of four Chinese NER datasets. We conducted a frequency analysis of characters appearing as entity starts and ends, sorted in descending order. Based on the analysis, we sum up the top 5% (10%) characters' frequency as entity start or end and divide it by the total frequency to get frequency ratio as $r^s_{5\%}$, $r^e_{5\%}$, $r^s_{10\%}$ and $r^e_{10\%}$.

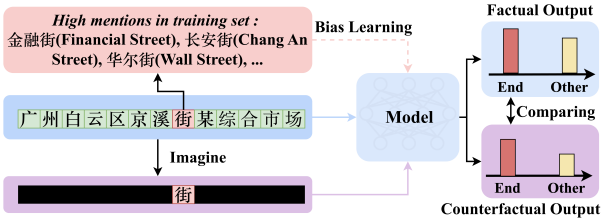

Figure 1: Illustration of counterfactual rethinking on boundary detection. The characters in green squares are the components of entity. The black square denotes mask operation in textual context.

As shown in Figure 1, when predicting the character "街 (street)" in sentence "广州白云区京溪街某综合市场 (A comprehensive market in Jingxi Street, Baiyun District, Guangzhou.)", the model wrongly identifies it as an entity end because of the learned ***boundary bias*** caused by high mentions of "街 (street)" as entity boundaries. Through counterfactual rethinking, the model imagines a counterfactual situation of only seeing the character mention and compares the factual output with the counterfactual, then makes decisions collectively based on textual context and character mentions. Thus, the model can focus on the main effect of the textual context while not losing the character features.

In our work, to study and mitigate the boundary bias while enhancing boundary supervision and span representation by multi-task learning in Chinese NER tasks, we propose SEN, a Span Enhanced Two-Stage Network with a boundary detector and a convolution-based span classifier for Chinese NER, and apply the Counterfactual Rethinking (***CR***) strategy on SEN in inference. Specifically, in the first stage of training, we utilize a boundary detector to learn entity boundary features for boundary supervision, and then employ a convolution-based classifier to encode N-gram features into span presentation and classify spans into corresponding categories in the second stage. Both modules utilize the same contextual embedding and are trained within a multi-task learning architecture using biased training data. In the inference stage, we first construct a causal graph for boundary detection to analyze the dependencies between variables, which acts as a "guidance" for capturing the causal effects of textual context and character mention. Then, to avoid the test instances being poisoned by boundary bias, our boundary detector "imagines" the counterfactual counterparts on our causal graph to distill the biases through counterfactual rethinking. Finally, we perform a bias removal operation to produce a counterfactual prediction that corresponds to a debiased decision.

We highlight that the proposed SEN with the CR strategy (SENCR) is lexicon-free and the CR strategy will not change any parameters of the trained model. To verify, we perform extensive experiments on four public Chinese NER benchmark datasets. The overall results demonstrate that SENCR achieves new state-of-the-art performance compared to recent competitive lexicon-free models, and even outperforms some existing lexicon-based models. Moreover, the improvements of F1-score (F1) on new boundaries of SENCR compared with SEN are 0.84%, 1.40%, 1.87% and 1.12% respectively in test sets of four datasets, that shows the effectiveness of the proposed CR strategy.

Related Work

Lexicon-Based Chinese NER. In Chinese NER, recent studies adopt lexicons to enhance boundary and semantic features in character-level representation. Zhang and Yang (2018) introduced a lattice LSTM structure to encode all characters and potential words recognized by a lexicon in a sentence, avoiding the error propagation of segmentation while leveraging the word information. CNN (Gui et al. 2019a) and GNN (Gui et al. 2019b) models were also employed to leverage better lexicon-based information. To fully utilize the parallel computation of GPUs, Li et al. (2020) introduced a flat-lattice transformer architecture to encode lexicon features. Moreover, Liu et al. (2021) proposed a novel method to integrate external lexicon knowledge into BERT layers for Chinese sequence labelling. However, the effectiveness of these aforementioned approaches heavily hinges on the quality of external lexicons.
Two-Stage NER. Two-stage NER refers to a process in which the identification and classification of named entities is performed in two separate stages. The first stage involves detecting and labeling the entity spans in the text, while the second stage involves assigning the appropriate categories to the detected spans. Zheng et al. (2019) combines sequence labeling model and region classification model to locate and classify nested entities with high performance. Tan et al. (2020) first predict boundary and then perform classification over span features. Wu et al. (2022) presents a novel

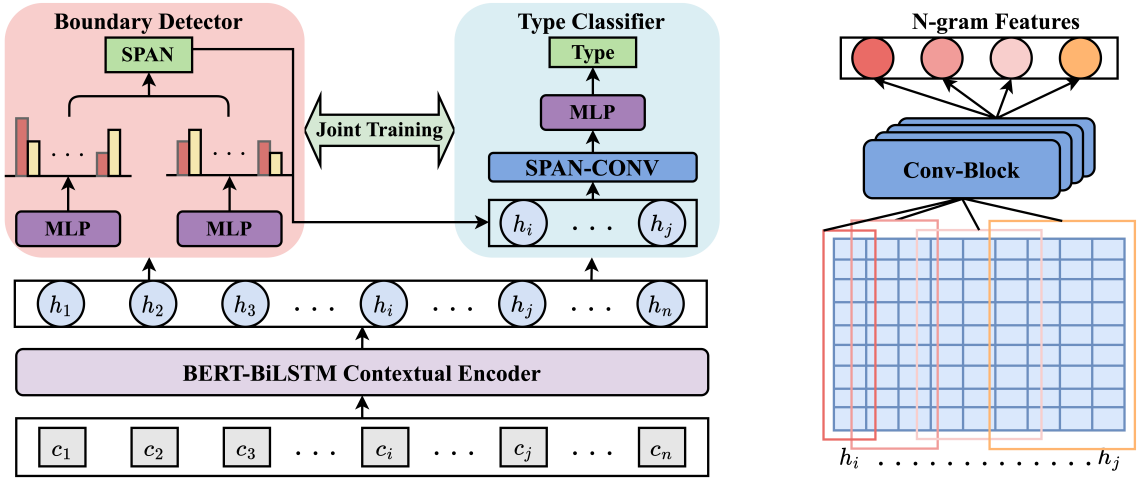

Figure 2: Overall architecture of SEN. The left part is the whole structure of SEN, which contains a boundary detection module and a type classification module. The right part illustrates the convolution operation of our SPAN-CONV block on the span.

two-stage set prediction network named Propose-and-Refine Network for nested English NER. Shen et al. (2021) treat NER as a joint task of boundary regression and span classification and propose a two-stage entity identifier. The main differences between our network and the above methods are that our model consists of a convolution-based neural network to capture different n-gram features for span enhancement and a novel CR strategy for debiased boundary prediction.

Causal Inference Causal inference (Pearl 2009; Pearl and Mackenzie 2018) aims to determine the independent, actual effect of a particular phenomenon, which has been employed in psychology, politics and epidemiology for many years (Sobel 1995; Richiardi, Bellocco, and Zugna 2013; Keele 2015). By removing confounding bias in data, causal inference can provide debiased solutions through estimating the causal effect rather than correlation. Inspired by causal inference, Liu et al. (2022) proposed a causal context debiasing recognition framework to remove the effect of contextual bias in vision recognition. Lin et al. (2022) utilized a new causal debiasing framework to eradicate the detrimental contrast distribution bias and spatial distribution bias in Unsupervised Salient Object Detection (USOD). In NLP, Qian et al. (2021) designed a counterfactual framework for text classification debiasing. A counterfactual analysis based method is proposed by Wang et al. (2022) to debias Relation Extraction (RE). Tian et al. (2022) propose a novel bias mitigation strategy to reduce known biases learned by Natural Language Understanding (NLU) models based on causal inference. Zhang et al. (2021) proposed a framework to identify and resolve the dictionary bias in Distant-Supervised NER via causal intervention. Inspired by these applications of causal inference, we aim to teach our model a debiased prediction by distinguish the main effects between the character mention and textual context in entity boundary detection through our proposed CR strategy.

Method

Span Enhanced Two-Stage Network

In this section, we introduce the proposed Span Enhanced Two-Stage Network (SEN) in details, as illustrated in Figure 2. We initially acquire character-level contextual embeddings using a conventional lexicon-free BERT-BiLSTM encoder. In order to achieve the goal of boundary supervision and span enhancement, we then construct SEN with two modules, including a boundary detector for boundary detection and a type classifier for span classification. Finally, we jointly train our SEN under a multi-task learning architecture to benefit the both two modules.

Character-Level Encoder. Character-level encoder is used to map discrete characters into continuous input vectors. Considering a Chinese sentence $s = \{c_1, c_2, ..., c_n\} \in V_c$ where V_c is the character vocabulary, we map each character into a real-valued embedding to represent its semantic and syntactic meaning. Each character c_i is encoded as:

$$x_i = \mathrm{BERT}(c_i) \tag{1}$$

where BERT denotes a standard BERT (Devlin et al. 2019) encoder. Then, the sequence of character embeddings will be fed to the bidirectional LSTM (BiLSTM) layer to get the character-level contextual embeddings as follows:

$$\overrightarrow{h_i} = \overrightarrow{\mathrm{LSTM}}(x_i, \overrightarrow{h_{i-1}}) \tag{2}$$

$$\overleftarrow{h_i} = \overleftarrow{\mathrm{LSTM}}(x_i, \overleftarrow{h_{i-1}}) \tag{3}$$

$$h_i = [\overrightarrow{h_i}; \overleftarrow{h_i}] \tag{4}$$

where $[;]$ denotes concatenation, and the final character-level sequence representation can be expressed as $H = \{h_1, h_2, ..., h_n\}$.

Entity Boundary Detector. Instead of classifying characters into B (Begin), M (Middle), E (End), S (Single), O (Other) to denote their roles in entities, our Entity Boundary Detector aims to predict whether a character in the sentence is the start or end of an entity. Specifically, we design

two task-oriented multi-layer perceptron (MLP) classifiers to predict the entity start and end positions.

In order to get the probability of the character c_i being the start or end of an entity, we employ layer normalization (Ba, Kiros, and Hinton 2016) to process the character representation h_i, and then feed it into the MLP classifier and apply a softmax layer on the output.

$$\hat{y}_i^s = \text{softmax}(\text{MLP}_\text{s}(\text{LayerNorm}((h_i)))) \quad (5)$$

$$\hat{y}_i^e = \text{softmax}(\text{MLP}_\text{e}(\text{LayerNorm}((h_i)))) \quad (6)$$

After obtaining the start probability $\hat{y}_i^s$ and the end probability $\hat{y}_i^e$ of character c_i, we apply a argmax function to get the final prediction $c_i^s, c_i^e \in \{0, 1\}$, where 0 represents that c_i is not an entity start or end, and 1 denotes that c_i is an entity start or end. Finally, we get the sequential start prediction $s^s = \{c_1^s, c_2^s, ..., c_n^s\}$ and end prediction $s^e = \{c_1^e, c_2^e, ..., c_n^e\}$.

Entity Type Classifier. In order to classify spans into corresponding categories, we design a span-specific Entity Type Classifier to leverage span features and predict the tags. Specially, we add a additional *NoneType* category to denote that the span is not belong to any named entity types.

Instead of enumerating all possible spans from the raw sentence, for every positive start prediction in s^s, we match its corresponding end pair by simply searching the nearest positive end prediction in s^e to get the final span boundary (i, j). Inspired by TextCNN (Kim 2014), we propose a convolution-based neural network to enhance span representations by extracting $k\text{-}gram$ features from spans. As illustrated in the right part of Figure 2, we employ multiple convolutional networks with different kernel sizes to capture multi-scale span representations as follows:

$$k\text{-}gram = MaxPooling(Conv_k(h_{i:j}) \quad (7)$$

where k denotes the kernel size of convolutional network, $h_{i:j}$ denotes the span character representation extracted by the boundary (i, j).

Finally, we concatenate n-gram features leveraged by our span enhanced convolutional network to obtain the final representation of the span r_{sp}, then feed it into a multi-layer perceptron classifier, and apply a softmax layer to obtain the probability $\hat{y}^{sp}$ of the corresponding category.

$$r_{sp} = [1\text{-}gram; 2\text{-}gram; ...; k\text{-}gram] \quad (8)$$

$$\hat{y}^{sp} = \text{softmax}(\text{MLP}_\text{sp}(\text{LayerNorm}(\text{r}_\text{sp})) \quad (9)$$

Joint Training. During training, we jointly minimize the cross-entropy loss of two subtasks in a multi-task learning form. For the boundary detection loss, we apply a binary cross-entropy loss to calculate the start loss $\mathcal{L}_b^s$ and end loss $\mathcal{L}_b^e$ respectively.

$$\mathcal{L}_b^s = -\sum_{i=1}^{n}[y_i^s log(\hat{y}_i^s) + (1 - y_i^s) log(1 - \hat{y}_i^s)] \quad (10)$$

$$\mathcal{L}_b^e = -\sum_{i=1}^{n}[y_i^e log(\hat{y}_i^e) + (1 - y_i^e) log(1 - \hat{y}_i^e)] \quad (11)$$

$$\mathcal{L}_b = \frac{1}{2}(\mathcal{L}_b^s + \mathcal{L}_b^e) \quad (12)$$

where y_i^s and y_i^e are the golden label indicating whether the $i\text{-}th$ character is the start or end of an entity or not. The loss function of the entity type classifier is defined as cross-entropy:

$$\mathcal{L}_{sp} = -\sum y^{sp} log(\hat{y}^{sp}) \quad (13)$$

where y^{sp} denotes the ground truth of span type. Finally, we define the total training loss $\mathcal{L}$ of SEN as below:

$$\mathcal{L} = \lambda \mathcal{L}_b + (1 - \lambda)\mathcal{L}_{sp} \quad (14)$$

where λ is the hyper-parameter that balances two subtasks.

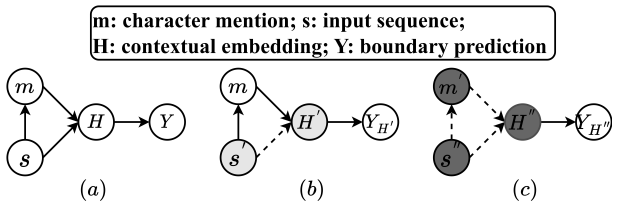

Figure 3: Causal graphs for boundary detector: (a). the original causal graph of boundary detection, (b). the counterfactual rethinking variant on boundary bias, (c). the counterfactual rethinking variant on dataset bias. The shading denotes the mask of corresponding variables.

Counterfactual Rethinking Strategy

In this section, we study and debias the boundary bias learned by our trained boundary detector using the CR strategy. We firstly formulate a causal graph for the boundary detection task from a causal perspective and then introduce the CR strategy to distill the boundary bias. Finally, we remove the distilled bias from the predictions of our boundary detector to improve its generalization.

Causal Graph for Boundary Detection. In order to implement the CR strategy, we first construct the causal graph (Pearl 2009; Pearl and Mackenzie 2018) for boundary detection in the inference stage, as shown in Figure 3(a). Causal graph is expressed visually by using directed acyclic graphs (DAGs), whose vertices are random variables and directed edges represent direct causation from variable $\mathcal{A}$ to variable $\mathcal{B}$. This graph reveals how the character mention and its context influence the contextual embedding of each word in the sequence, thereby impacting the prediction of boundaries.

Concretely in Figure 3(a), (1) the causal link $s \rightarrow m$: the character mention is determined by the input sequence; (2) the causal link $s \rightarrow H$: in inference, contextual embeddings are encoded from input sequence by our trained BERT-BiLSTM encoder; (3) the causal link $m \rightarrow H$: Also, character mention is inevitably encoded into contextual embeddings by the trained encoder; (4) the causal link $H \rightarrow Y$: the boundary detector utilizes contextual embeddings to get boundary prediction; (5) the causal link $m \rightarrow H \rightarrow Y$: the boundary detector is misled by character mention encoded in contextual embeddings and get the biased boundary prediction; (6) the causal link $s \rightarrow H \rightarrow Y$: both characters

and its context are comprehensive understood and encoded in contextual embeddings and the boundary detector gets the fair prediction of each character.

Bias Distillation. Based on the causal links in Figure 3(a), we analyze how the boundary bias affects boundary detector in inference. In the inference stage, the learned model parameters indicates causal dependencies among the variables. The boundary bias hurts the model generalization to make wrong predictions mainly through the causal link $m \rightarrow H \rightarrow Y$ while ignoring the real main causal link $s \rightarrow H \rightarrow Y$. Thanks to causal inference, we no longer treat the whole inference process as a black box. In contrast, we utilize the causal intervention which is denoted as *do*() to realize our CR strategy on the trained boundary detector by manipulating the nodes and observe the new output.

To distill the boundary bias by counterfactual rethinking, we apply *do-operation* on s which wipes out all the incoming links of s and alter it to the counterfactual. Here, we mask the textual context, but maintain m as the original to get the output logits $Y_{H'}$(see Figure 3(b))as follows:

$$Y_{H'} = Y(do(s = s^{'})) \tag{15}$$

$H^{'}$ denotes the contextual embeddings after intervention $s = s^{'}$, the original prediction can be denote as Y_H.

By masking textual context in s, the character will not indicate any boundary preference in sentence. In this case, since the model cannot see any textual context in the fact input s after the invention $s = s^{'}$, but still has access to the original character mention m as the inputs, the prediction $Y_{H'}$ purely reflects the side effect from m. In other words, $Y_{H'}$ refers to the output affected by ***boundary bias***, where only the character mentions are available as input while textual context is masked.

In addition to $Y_{H'}$ that reflects the side causal effects of character mentions, there is another kind of bias not conditioned on the character mentions m, but reflecting the general bias in the whole dataset caused by its collection and annotation procedure, which is $Y_{H''}$. The output $Y_{H''}$ corresponds to the counterfactual input sequence s where both textual context and character mentions are removed. In this case, since the model cannot see any information from the input sequence $s^{''}$ after intervention $s = s^{''}$(see Figure 3(c)), $Y_{H''}$ naturally reflects the dataset bias that was learned by the trained model in training.

Bias Removal. Our final goal is to use the direct effect from H to Y for debiased boundary prediction while removing the boundary bias and dataset bias learned by our boundary detector in biased training. The debiased prediction via bias removal can be formalized via the conceptually simple and empirically powerful element-wise subtraction operation:

$$Y_{debiased} = Y_H - \alpha_1 Y_{H'} - \alpha_2 Y_{H''} \tag{16}$$

where $Y_{debiased}$ is the final debiased boundary prediction; $Y_{H'}$ and $Y_{H''}$ correspond to the boundary bias and dataset bias distilled from the trained boundary detector, respectively; α_1 and α_2 are two independent hyper-parameters balancing the two types of biases. We adaptively set the values of α_1 and α_2 for different datasets using grid search in a scoped two dimensional space:

$$\alpha_1^*, \alpha_2^* = arg \max_{\alpha_1, \alpha_2} \psi(\alpha_1, \alpha_2) \quad \alpha_1, \alpha_2 \in [\hat{a}, \hat{b}] \tag{17}$$

where ψ is the F1 metric, $\hat{a}, \hat{b}$ are the boundaries of the search range. The two hyper-parameters are at dataset-level and thus searched only once for each validation set, and would be used in inference all testing instances.

Experiments

To evaluate the performance of the proposed SEN and the CR strategy in inference, we conducted extensive experiments on four Chinese NER datasets covering different domains. In this section, we describe the details of the four datasets, implementation settings, main results, ablation study, F1 against N-gram selections and analysis on CR strategy in the experiments.

Datasets

CLUENER (Xu et al. 2020). It is a well-defined fine-grained dataset for Chinese NER collected from Sina News. Its entities contain 10 different categories, including organization, person name, address, company, government, book, game, movie, position, and scene.

MSRA (Levow 2006). MSRA is also a dataset annotated from news domain and contains 3 types of named entities: LOC (Location), PER (Person) and ORG (Organization).

Resume (Zhang and Yang 2018). Resume dataset is composed of resumes from Sina Finance and is annotated with 8 types of named entities: LOC, PER, ORG, CONT (Country), EDU (Educational Institution), PRO (Profession), RACE (Ethnicity) and TITLE (Job Title).

Weibo (Peng and Dredze 2015). Weibo social dataset contains 4 different categories, including LOC, PER, ORG, GPE (Geo-Political Entity).

We followed the same training, development, test split on Weibo and Resume datasets as Li et al. (2020). Development set and test set are not available for MSRA dataset and CLUENER2020 dataset respectively, we followed Gui et al. (2019b) to use test set or dev set instead.

Implementation Details

Our proposed SENCR is implemented with the Pytorch framework. For the BERT encoder, we utilize the standard pre-trained Chinese BERT-base model with 768-dimensional hidden representations to acquire character embeddings, completely free from any outer lexicons. As for hyper-parameter configurations, we search for the best hyper-parameters in development set and evaluate on test set to obtain the final performance for different datasets respectively. The hidden states of the BiLSTM and span-convolution neural network are set to 256 and 200 for Resume and Weibo datasets, 200 and 100 for CLUENER dataset, and 300 and 200 for MSRA dataset, respectively. To avoid overfitting, we employ a dropout rate of 0.1 on the BiLSTM, convolution-based neural network, and MLPs for the Resume and MSRA datasets. For the Weibo and

Models	Weibo	Resume	MSRA	CLUENER
BERT+FLAT(2020)	68.55	95.86	96.09	-
SoftLexicon(2019)	70.50	96.11	95.42	-
LEBERT(2021)	70.75	96.08	95.70	-
MECT(2021)	70.43	95.98	96.24	-
DCSAN(2021)	71.27	96.67	**96.41**	-
MCL(2023)	73.08	96.46	96.11	-
BERT+CRF*	67.33	95.51	94.83	79.65
BERT+GAM*(2022a)	63.60	-	94.97	81.08
W^{2}NER*(2022b)	72.32	96.65	96.08	-
SEN(base)	72.73	96.86	96.12	81.61
SENCR	**73.42**	**97.13**	96.17	**81.86**

Table 2: Main Results(F1) on Resume, MSRA, Weibo and CLUENER datasets. * denotes model with no lexicons. BERT+FLAT (Li et al. 2020), SoftLexicon (Ma et al. 2019), LEBERT (Liu et al. 2021), MECT (Wu, Song, and Feng 2021), DCSAN (Zhao et al. 2021), MCL (Zhao et al. 2023), BERT+GAM (Li et al. 2022a), W^{2}NER (Li et al. 2022b).

CLUENER datasets, the dropout rate is set to 0.5. To train the model, we use AdamW (Loshchilov and Hutter 2017) optimizer with a learning rate of 1e-5 for fine-tuning BERT and a learning rate of 1e-3 for other part of the proposed SEN. In addition, we randomly sample part of negative spans detected from the boundary detector as the additional *NoneType* for data augmentation to the entity type classifier. For the CR strategy, we search the best α_1 and α_2 in development set with range [-2, 2] and step 0.02. For evaluation, Standard Precision (P), Recall (R), and F1 are employed as evaluation metrics for both boundary and entity. All of our experiments are conducted on the same machine with two Nvidia RTX 3090 GPUs.

Overall Performance

We present the overall results on four Chinese NER datasets in Table 2. As shown in this table, we can observe that our proposed SEN achieves the state-of-the-art performance compared with lexicon-free models on these datasets. Moreover, SEN with CR strategy (SENCR) even outperforms recent lexicon-based methods on Weibo and Resume. Concretely, on Weibo, SEN achieves 0.41 absolute F1 improvement over the lexicon-free method W^{2}NER and with CR strategy integrated, the F1 improves 0.69 and surpasses all the recent methods. On Resume, SEN obtains a improvement compared to the SOTA approach DCSAN by 0.19 F1 and a decent improvement of 0.27 F1 after applying CR strategy. On MSRA, although the improvement of SEN and SENCR over the lexicon-free SOTA model W^{2}NER is limited, our models still surpass some of the lexicon-based models such as BERT-FLAT, SoftLexicon, LEBERT and MCL. In addition, on CLUENER, we can only find two lexicon-free models for comparison since CLUENER is released recently. Compared with BERT+GAM, SEN outperforms it by 0.53 F1 and with the CR strategy, the improvements is 0.25 F1. The above results demonstrates that our proposed lexicon-free SEN can achieve comparable performance by boundary supervision and span enhancement under a multi-task training architecture. Moreover, the improvements of SENCR compared to SEN on four datasets show the effectiveness of our proposed ***CR*** strategy.

Ablation Study

We conduct the ablation study in the following three aspects on Resume and CLUENER datasets, as shown in Table 3.

Span enhancement. To validate the effectiveness of our proposed convolution-based network for span enhancement, we replace this block with the concatenation of start and end tokens' representation and the span length embedding. The F1 drops by -0.60 and -0.92 on Resume and CLUENER datasets without span enhancement. It shows that our proposed convolution-based network can leverage richer semantic features by n-gram convolutions than traditional concatenation method.

NoneType augmentation. In the ablation experiment without data augmentation on our entity classifier, we do not sample any negative spans from the boundary detector. The performance drops by -0.49 and -0.64 on Resume and CLUENER datasets compared with random sampling. This indicates random sampling negative spans from boundary detection as additional *NoneType* can enhance robustness of the entity classifier and endow the classifier to distinguish negative spans originating from the upstream task.

Multi-task joint training. In the ablation experiment without multi-task joint training, we train the two-stage model SEN in a pipeline manner. The performance drops significantly by -2.36 and -3.33 on Resume and CLUENER datasets. This demonstrates the substantial mutual benefits that can be derived from the interplay between boundary detection and entity classification tasks within a multi-task joint training architecture.

Model	Resume			CLUENER		
	P	R	F1	P	R	F1
SEN(base)	**97.16**	**96.56**	**96.86**	**81.97**	**81.25**	**81.61**
- SE.	96.26	96.26	96.26	81.92	79.49	80.69
- DA.	96.51	96.22	96.37	81.86	80.08	80.97
- JT.	94.35	94.64	94.50	77.35	79.42	78.28

Table 3: Ablation Study. SE denotes span enhancement, DA means *NoneType* data augmentation on our entity classifier and JT is the multi-task joint training.

F1 against N-gram Selections

To analyze the impact of N-gram selections on model performance, we conduct experiments on CLUENER and Resume datasets by employing our convolution-based network with different kernel size lists. As shown in Figure 4, SEN with kernel size list of 1-4: $[1, 2, 3, 4]$ and 1-5: $[1, 2, 3, 4, 5]$ achieves the best F1 on CLUENER and Resume datasets respectively. The results above indicate that leveraging n-gram features to enhance span representations with our convolution-based network can improve the performance of SEN to a certain extent. However, it also suggests that

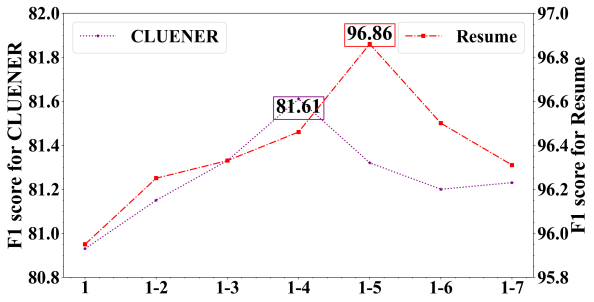

Figure 4: F1 against N-gram selections on CLUENER and Resume datasets. 1-n denotes span-convolution network with kernel size list: 1, 2, ... , n.

Figure 5: Case Study on CLUENER dataset. In the figure, the green characters are the golden and the correctly identified entities and boundaries. The red characters are the wrongly identified items.

the model's performance does not necessarily improve with more N-grams. This observation can be attributed to the fact that N-grams with longer lengths than spans may introduce noise rather than conveying meaningful semantic features.

Analysis on CR Strategy

Our proposed CR strategy aims to distill and remove bias caused by boundary character with high mention frequency so as to improve the generalization ability of boundary detector in inference stage. To verify the effectiveness of CR strategy on boundary detection, we evaluated and compared the performance of SEN and SENCR on new boundaries in test sets of four Chinese datasets. As shown in Table 4, after applying our CR strategy on boundary detector of SEN, the F1 of new boundaries improves 0.84, 1.40, 1.87 and 1.12 on Resume, CLUENER, Weibo and MSRA datasets respectively. Moreover, we analyze two cases from the CLUENER test set, as shown in Figure 5. In the first case, character "街 (Street)" mislead the boundary detector of SEN to identify it as end of an entity because of "街 (Street)" is usually appears as entity ends in training set such as "长安街 (Chang An Street)", "美食街 (Food Street)" and so on. In the second case, the highly mentioned special character '》' is regarded as entity end for thousands of times in training set that guide the boundary detector to simply remember it as boundaries. As a result, SEN wrongly identified "广州白云区京溪街 (Jingxi Street, Baiyun District, Guangzhou)" and "《黑光》(Blacklight)" as final entity predictions while ignored the textual context "某综合市场 (a comprehensive market)" and "（Blacklight）". With CR strategy, SENCR can correctly identify "广州白云区京溪街某综合市场 (a comprehensive market in Jingxi Street, Baiyun District, Guangzhou)" and "《黑光》（Blacklight）(Blacklight)" as debiased prediction by removing the negative impact of ***boundary bias*** in inference. The improvements of F1 on new boundaries and the case study on CLUENER dataset indicate that the CR strategy can help the boundary detector achieve higher generalization by mitigating the ***boundary bias*** learned in the training procedure.

Model	Resume		CLUENER	
	All	New	All	New
SEN(base)	98.02	96.20	89.08	67.13
SENCR	98.41	**97.04**	89.54	**68.53**

Model	Weibo		MSRA	
	All	New	All	New
SEN(base)	77.68	61.65	97.31	88.50
SENCR	78.40	**63.52**	97.42	**89.62**

Table 4: F1 for all and new boundaries in test sets of four Chinese datasets. All denotes all boundaries in the test set. New denotes boundaries in the test set that have not been seen in the training set.

Conclusion

In this paper, we proposed a lexicon-free Chinese NER framework called SENCR that incorporates a boundary detector for boundary supervision, a span-convolutional network for better span representation and classification and a novel counterfactual rethinking strategy in inference for debiased boundary detection. The experimental results on four Chinese NER datasets show that SENCR achieves state-of-the-art performance compared to other lexicon-free approaches on four datasets and even outperformed lexicon-based approaches on Resume, Weibo and CLUENER datasets. In addition, evaluation on new boundaries in test sets of four Chinese datasets proves the effectiveness of our proposed CR strategy.

Acknowledgments

This work was supported by grants from the National Major Science and Technology Projects of China (grant no. 2022YFB3303302), the National Natural Science Foundation of China (grant nos. 62377040, 62207007, 61977012).

References

Ba, J.; Kiros, J. R.; and Hinton, G. E. 2016. Layer Normalization. *ArXiv*, abs/1607.06450.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv*, abs/1810.04805.

Gui, T.; Ma, R.; Zhang, Q.; Zhao, L.; Jiang, Y.-G.; and Huang, X. 2019a. CNN-Based Chinese NER with Lexicon Rethinking. In *ijcai*, volume 2019.

Gui, T.; Zou, Y.; Zhang, Q.; Peng, M.; Fu, J.; Wei, Z.; and Huang, X. 2019b. A Lexicon-Based Graph Neural Network for Chinese NER. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 1039–1049.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Philip, S. Y. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2): 494–514.

Keele, L. 2015. The Statistics of Causal Inference: A View from Political Methodology. *Political Analysis*, 23(3): 313–335. Publisher: Cambridge University Press.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Conference on Empirical Methods in Natural Language Processing*.

Levow, G.-A. 2006. The Third International Chinese Language Processing Bakeoff: Word Segmentation and Named Entity Recognition. In *SIGHAN@COLING/ACL*.

Li, D.; Yan, L.; Yang, J.; and Ma, Z. 2022a. Dependency syntax guided bert-bilstm-gam-crf for chinese ner. *Expert Systems with Applications*, 196: 116682.

Li, J.; Fei, H.; Liu, J.; Wu, S.; Zhang, M.; Teng, C.; Ji, D.; and Li, F. 2022b. Unified Named Entity Recognition as Word-Word Relation Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10): 10965–10973.

Li, X.; Yan, H.; Qiu, X.; and Huang, X. 2020. FLAT: Chinese NER Using Flat-Lattice Transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6836–6842. Online: Association for Computational Linguistics.

Liang, G.; and Leung, C. W.-K. 2021. Improving model generalization: A Chinese named entity recognition case study. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 992–997.

Lin, H.; Lu, Y.; Tang, J.; Han, X.; Sun, L.; Wei, Z.; and Yuan, N. J. 2020. A Rigorous Study on Named Entity Recognition: Can Fine-tuning Pretrained Model Lead to the Promised Land? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7291–7300. Online: Association for Computational Linguistics.

Lin, X.; Wu, Z.; Chen, G.; Li, G.; and Yu, Y. 2022. A Causal Debiasing Framework for Unsupervised Salient Object Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2): 1610–1619.

Liu, R.; Liu, H.; Li, G.; Hou, H.; Yu, T.; and Yang, T. 2022. Contextual Debiasing for Visual Recognition with Causal Mechanisms. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12745–12755. New Orleans, LA, USA: IEEE. ISBN 978-1-66546-946-3.

Liu, W.; Fu, X.; Zhang, Y.; and Xiao, W. 2021. Lexicon enhanced Chinese sequence labeling using BERT adapter. *arXiv preprint arXiv:2105.07148*.

Loshchilov, I.; and Hutter, F. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Ma, R.; Peng, M.; Zhang, Q.; and Huang, X. 2019. Simplify the usage of lexicon in Chinese NER. *arXiv preprint arXiv:1908.05969*.

Miwa, M.; and Bansal, M. 2016. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1105–1116. Berlin, Germany: Association for Computational Linguistics.

Mollá, D.; Van Zaanen, M.; and Smith, D. 2006. Named entity recognition for question answering. In *Proceedings of the Australasian language technology workshop 2006*, 51–58.

Pearl, J. 2009. *Causality*. Cambridge University Press. ISBN 978-1-139-64398-6.

Pearl, J. 2021. 7.1 causal and counterfactual inference. *The handbook of rationality*, 427.

Pearl, J.; and Mackenzie, D. 2018. *The Book of Why: The New Science of Cause and Effect*. Penguin Books Limited. ISBN 978-0-241-24264-3.

Peng, N.; and Dredze, M. 2015. Named Entity Recognition for Chinese Social Media with Jointly Trained Embeddings. In *Conference on Empirical Methods in Natural Language Processing*.

Qian, C.; Feng, F.; Wen, L.; Ma, C.; and Xie, P. 2021. Counterfactual Inference for Text Classification Debiasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 5434–5445. Online: Association for Computational Linguistics.

Richiardi, L.; Bellocco, R.; and Zugna, D. 2013. Mediation analysis in epidemiology: methods, interpretation and bias. *International Journal of Epidemiology*, 42(5): 1511–1519. _eprint: https://academic.oup.com/ije/article-pdf/42/5/1511/1712729/dyt127.pdf.

Shen, Y.; Ma, X.; Tan, Z.; Zhang, S.; Wang, W.; and Lu, W. 2021. Locate and Label: A Two-stage Identifier for Nested

Named Entity Recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2782–2794. Online: Association for Computational Linguistics.

Sobel, M. E. 1995. Causal Inference in the Social and Behavioral Sciences. In Arminger, G.; Clogg, C. C.; and Sobel, M. E., eds., *Handbook of Statistical Modeling for the Social and Behavioral Sciences*, 1–38. Boston, MA: Springer US. ISBN 978-1-4899-1292-3.

Tan, C.; Qiu, W.; Chen, M.; Wang, R.; and Huang, F. 2020. Boundary Enhanced Neural Span Classification for Nested Named Entity Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05): 9016–9023.

Tian, B.; Cao, Y.; Zhang, Y.; and Xing, C. 2022. Debiasing NLU Models via Causal Intervention and Counterfactual Reasoning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10): 11376–11384.

Wang, Y.; Chen, M.; Zhou, W.; Cai, Y.; Liang, Y.; Liu, D.; Yang, B.; Liu, J.; and Hooi, B. 2022. Should We Rely on Entity Mentions for Relation Extraction? Debiasing Relation Extraction with Counterfactual Analysis. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 3071–3081. Seattle, United States: Association for Computational Linguistics.

Wu, S.; Shen, Y.; Tan, Z.; and Lu, W. 2022. Propose-and-Refine: A Two-Stage Set Prediction Network for Nested Named Entity Recognition. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 4418–4424. Vienna, Austria: International Joint Conferences on Artificial Intelligence Organization. ISBN 978-1-956792-00-3.

Wu, S.; Song, X.; and Feng, Z. 2021. MECT: Multi-metadata embedding based cross-transformer for Chinese named entity recognition. *arXiv preprint arXiv:2107.05418*.

Xu, L.; Dong, Q.; Yu, C.; Tian, Y.; Liu, W.; Li, L.; and Zhang, X. 2020. CLUENER2020: Fine-grained Name Entity Recognition for Chinese. *arXiv preprint arXiv:2001.04351*.

Zeng, X.; Li, Y.; Zhai, Y.; and Zhang, Y. 2020. Counterfactual generator: A weakly-supervised method for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7270–7280.

Zhang, W.; Lin, H.; Han, X.; and Sun, L. 2021. De-biasing Distantly Supervised Named Entity Recognition via Causal Intervention. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 4803–4813. Online: Association for Computational Linguistics.

Zhang, Y.; and Yang, J. 2018. Chinese NER Using Lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1554–1564. Melbourne, Australia: Association for Computational Linguistics.

Zhao, S.; Hu, M.; Cai, Z.; Chen, H.; and Liu, F. 2021. Dynamic Modeling Cross- and Self-Lattice Attention Network for Chinese NER. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16): 14515–14523.

Zhao, S.; Wang, C.; Hu, M.; Yan, T.; and Wang, M. 2023. MCL: Multi-Granularity Contrastive Learning Framework for Chinese NER. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11): 14011–14019.

Zheng, C.; Cai, Y.; Xu, J.; Leung, H.-f.; and Xu, G. 2019. A Boundary-aware Neural Model for Nested Named Entity Recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 357–366. Hong Kong, China: Association for Computational Linguistics.

Zhu, Y.; Wang, G.; and Karlsson, B. F. 2019. CAN-NER: Convolutional attention network for Chinese named entity recognition. *arXiv preprint arXiv:1904.02141*.